$1.50
MAY, 1985
VOL.5 NO.4

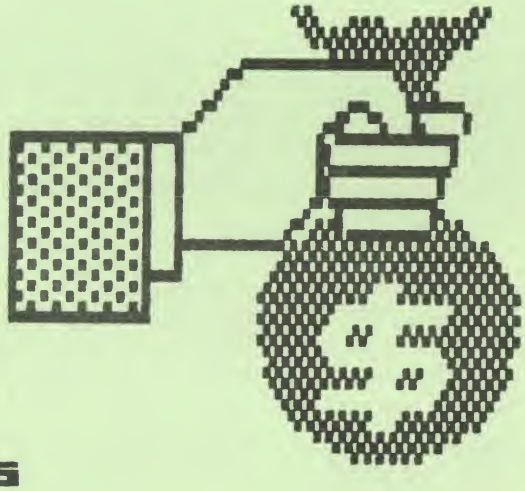
M.A.C.E. JOURNAL

"Devoted Exclusively To The Atari Computer User"

IN THIS ISSUE...

TAXES

by Bob Pettapiece



BBS Numbers

Book Reviews

Software Reviews

Best Books for the Atari

BASIC Error Messages Explained

Published by the Michigan Atari Computer Enthusiasts

FIRESIDE CHAT

Well, here we are folks, coming up on the big 5 year mark. Next month MACE will celebrate its 5th birthday! Just think - 5 years ago there was a handful of people in the back of a computer store and look where we are now! What other computer, besides Atari, can keep so many people interested for so long? Here's to 5 more!

Speaking of the Atari, the new ones should be arriving in the stores soon. By the time you read this article MACE should have in its possesion one of the new 128K XE machines. We found one and we bought it for $169.00, and hopefully, it's right here at this meeting. (Remember, I'm writing this article in advance!)

Data World gave me a call on Saturday, May 6th, and told me they had 2 of the XE's and I asked him to ship one to us. We should have it about a week before this meeting. By the time you start asking questions tonight this machine will have been tested with tons of various software, used over the phone, and used as the MACE WEST BBS. Hopefully any compatibility problems, if there are any, should have been found by now.

As far as the other new hardware goes, I don't know! Data World doesn't know. Our program director, Scott, did talk to Atari on the phone but couldn't, of course, get any type of commitment from them. They did say that while we can't get people from Atari we may be able to get equipment by the birthday meeting in May. Let's keep our fingers crossed.

Next month is the birthday party! It's the only meeting out of the whole year that you must be a member to attend. Membership cards will be checked, and new ones will be available at the door. I've got lots of good prizes, or presents, including a 1050 drive to give away so I hope I'll see you there!

Kirk

SUCH A DEAL...

MACE has a couple good offers this month (from local dealers); just walk in with your MACE card.

Do you still want the MPP 64K printer buffer (Micro Stuffer)? Yes, you say! Then drive on out to Sy Draft in Lincoln Center at 10 1/2 mile and Greenfield, Oak Park. He has them, in stock, for a special MACE group price. ($103.00)

If you're interested in the US Doubler chip for the Atari 1050 drive you can get one for $55.00 from Just Software in East Detroit. The folks at Just Software (Coming Attractions) are aware of the "bug" so don't worry about getting the wrong chip from them. (Don't get anything earlier than Rev. K.)

THANKS!

A special thanks to the following businesses for providing prizes for next months birthday meeting; The Family Computer Center, Rite-Way, Just Software, OSS, Sector One International, and Data World. So folks, when you're out doing your Atari shopping, remember who our friends are!

** *** ** **** ** *** **

The MACE Journal is published monthly by the Michigan Atari Computer Enthusiasts.

Submissions to the Journal can be mailed to the PO Box, uploaded to the MACE BBSs, any officer's BBS, the Supership at (313) 288-0824, or uploaded directly to the editor at 646-4455. Where possible, submissions should include a disk or tape file in AtariWriter or similar format and a working copy of the program. Specify format for screen dumps (AtariArtist, Koalapad, etc.). Authors whose submissions are published will receive a certificate good for a free disk or tape from the MACE library. Deadline for submissions is the first of each month.

UNDERSTANDING TAXES

by Bob Pettapiece

Introduction

After being audited twice by the I.R.S. and spending a lot of time sitting while the auditor wrote down my list of deductions, I decided to put my Atari to use to keep track of my deductions. That was in 1983. The original program only listed and sorted deductions; it now allows me to change an entry and also keeps track of my sources of income. I am at a point where the program seems complete, at least for now.

I have been programming in BASIC for ten years, first as a hobby and later as a subject to teach in school. I believe in learning through experience. The Taxes program has been just that, a learning experience. The program does three important things: it stores, sorts and displays my tax data on either the screen or on a printer. This article is an attempt to show why I wrote Taxes the way I did so others can learn from my experience. Also, one may look at my program, make further modifications and make it run better. One aspect that could be improved is the sort. It works, but it is slow.

If you want to read this article and not type in the program, it is available on many BBSs in the area as well in the MACE disk library. I use inverse video for REM statements. This makes it easier to find various procedures when you troubleshoot or modify a part of the program. Relevant program lines are listed before each comment section. The program is presented in a discussion format and many lines are discussed out of order. Some familiarity with BASIC programming is assumed.

Beginning of the Program
[10-30,2760-2780]

The first subprogram, lines 2770-2780, dimensions the string ($) variables for the file names and sets the line length of an entry to 80. This is two screen widths or one normal printer width. Very handy!! Line 2780 dimensions variables for deduction input. The reason this is a subprogram is because it is used more than once. A subroutine (subprogram) is used when a procedure may be needed more than once in a program. It not only saves time typing and memory space, but BASIC likes them and operates quickly even though they are in a different part of the program.

[40-70,2790-2810]

POKE 1790, YEAR in line 70 is used to preserve the two digits of the year you are working with so the variables can be redimensioned. This information is POKEd into a safe part of memory that is not affected by the run of the program and is recalled in line 2790.

Line 70 also sends us to a GOSUB which sets up file names. In lines 2790-2810 the names of the files used to store the information are put into variables D1$, D2$ & D3$; DEDUCTNS holds the itemized list of deductions, TAXSPAID holds the income information and ENTRYTOT holds only one number, the total of deductions. Notice how LEN(1,2) of each file name is "D:" and LEN(12,13) is where the year (DN$) is appended. This provides similar files for each tax year.

[80-140]

Lines 100-140 create new files if it is the first entry for that tax year. Line 140 stops the program if locked files are encountered. Note the use of TRAP 0. This works as well as TRAP 40000 and saves typing.

[150-210,2840-2850]

Lines 150-210 are a menu allowing you to choose which part of the program to work on. I use a GET rather than an INPUT command because it works faster, using only one keystroke. Also embedded in these lines are the housekeeping chore of setting variables used in the deduction totals to zero. If this is not done, a return to this section after doing totals will result in the wrong totals the next time through; the variables keep the old totals and add the new figures to them.

Also a part of the main menu is the option to finish the session. Included in the finish is the locking of files (XIO #35) and CLOSEing of channels which had been OPENed.

[220-330]

Line 220 unlocks & OPENs the files related to deductions and retrieves the number of entries stored. Storing the number of entries allowed me to write FOR...NEXT loops when reading data from a file or storing it to a file rather than depending on out-of-data errors and TRAP statements to do the job. Much neater this way! Lines 230-320 are the menu for deductions. The use of ON...GOTO makes programming much easier and shorter than a series of IF...THEN statements.

[340-360]

The categories set up in this program (lines 340-360) were for my convenience, even though they fit into a Form 1040 format. They can be changed to fit your needs.

[370-430]

Each part of a deduction is asked for separately. This allows a sort to occur on any part. The program only sorts the first two, category and check number.

[440-510]

Lines 440 & 450 double check the entry. If incorrect, the program simply starts the process of entry over again. If correct, the entry is stored at line 460 and ENTRIES is increased by one.

[520-570]

Lines 520-530 check for amounts that need trailing zeros. $5.4 in normal computer output needs a zero to look good, $5.40. The variable A is converted to a string, A$=STR$(A) in line 510 to make this possible.

[580-1070,2820-2830]

I originally wrote this program to work on an Epson printer. I later converted this back to generic printer commands. Note the use of DN$ in line 610 to print the year.

Line 600 looks for the printer to be turned on. M=1 is a flag to tell the routine where to return since there are several print routines that use TRAP 2830. The routine is included here so you can see how it works in this situation.

[1080-1160]

The following section is long, but it saves deleting whole entries and reentering them when only a part needs to be changed.

The following line OPENs the file "SCRATCH" to hold the data you have read from the deductions file and are keeping.

Here the computer is looking for the category you input (CATCOMP$). If it is the category (CAT$), the data is displayed, otherwise the data from the other categories are read into the file SCRATCH.
[1170-1350]

In lines 1170-1190 the old deductions file is DELETEd (XIO #33) and the file SCRATCH is RENAMEd (XIO #32) the deductions file, DEDUCTNS.

Sorting Deductions
[1350-1420]

This is where we sort out the list! First, line 1360 saves the number of entries for future reference. This is important if we are to redimension the variables. Why would we do such a silly thing?? The reason is, if we are to sort more than once during a run of the program we will have more or fewer entries if we add or delete some. As a consequence, it is easier simply to redimension string variables and start with a clean slate. We also will blank out (make equal to " ") the strings so any old information does not wander into our new sort.

Since both sorts are similar, I am only going to give details of the Category Sort.

[1430-1600,2860-2940]

In the routine starting at line 2860 many

things happen. Line 2890 stores the screen "turn on" value found in location 559 and saves it to the variable SCR. The location 559 is then POKEd with 0 to shut off the screen display. Why, you ask? The sorting business will go up to 30% faster if the computer does not have to maintain a screen display while shuffling our deductions. (It gets turned on again in line 1730.) In line 2900 we check to see if this is the first sort of the run from line 2940 which sets SORTED=1.

A moment here on computer logic. If a variable (SORTED) is equal to one (1) then it is true (IF SORTED THEN 2920); if equal to zero (0), then false. This makes a nice logical statement and it is shorter than IF SORTED=1 THEN 2920. The first time the program is run SORTED=0 and is false, therefore not jumping to line 2920. It instead continues to line 2910 and dimensions the variable C$ for the sort.

[2920-2930]

Lines 2920 & 2930 blank out the length of C$, 80 blanks at a time (line 2930). This is a refinement of the Bubble Sort in the Atari manual which did it one space at a time and it works much faster.

[1460-1770]

The same blanking out is done for TEMP$ which holds 80 spaces for each entry in line 1470. Each entry is entered as it was stored, with Category first since it is the key for this sort. The reason parts of an entry are started at various parts of TEMP$ (10,41,etc.) is to put the output in columns. The starting spot for A (A$) is determined in line 1560, placing it on the extreme right of TEMP$.

[1770-1910]

Line 1780 POKEs the screen (82 & 83) to display 40 characters/line; two lines equal one full entry. Line 1820 returns the screen display back to normal.

[1920-2100]

Here is the second sort, by check number. The only real difference is in lines 1990 & 2000 where TEMP$ has the check number (CK$) stored before the category (CAT$) in order to sort on that item.

I also have a built in word BILL when "R" is entered for the check number. I also use MILE for mileage expenses and TABL for tax table entries. Since CK$ is a string I can use any four character entry besides a number. They get printed at the end, after check numbers since their ASCII code is higher.

Notice at line 2100 the program is sent to 1610 to use the same Bubble Sort routine. Once the information is stored in lengths of 80 characters, the sort is run the same way.

[2110-2140]

Once the sorting is done, CLR in line 2110 zaps all variables and they are redimensioned by the GOSUBs, ready to sort again if called upon.

Sources of Income
[2150-2750]

The procedures and routines in this section are fairly easy to understand if you have worked your way this far. I did not go to the trouble with this section as with deductions because of my few sources of income. As a result, one could do many of the types of things with this section as was done with deductions; such as sorts and changing parts of an entry.

Hints
[2950-2960]

This last section could go at the beginning of the program as well as the end. Since I use this program only once a year, I include hints for myself as well as for others. Being at the end of the program I can add or delete ideas as I see fit.

Conclusion

There you have it, your own Taxes program and hints to further programming. I hope you learned a little, improved upon my program a little and have a program you can use and enjoy. I would be glad to discuss modifications at any MACE meeting.

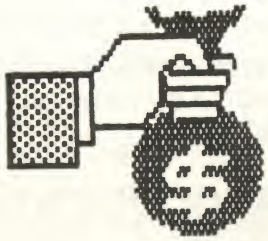
```
10 REM Program by BOB PETTAPIECE, 1983
20 REM **** Revised 1985 ****
30 GOSUB 2770
40 REM File Set-up
50 OPEN #3,12,0,"K:"
60 GRAPHICS 2:POSITION 1,0:? #6;"WHAT
ARE THE LAST 2DIGITS OF THIS  TAX YEAR
 (19??)?":INPUT YEAR
70 POKE 1790,YEAR:GOSUB 2790
80 POSITION 1,3:? #6;"is this your fir
st entry for this tax  year?":GET #3,B
:IF B<>89 THEN 150
90 ? #6;"ARE YOU SURE??":GET #3,B:IF B
<>89 THEN 150
100 TRAP 140
110 OPEN #1,8,0,D1$:CLOSE #1
120 OPEN #2,8,0,D2$:CLOSE #2
130 OPEN #4,8,0,D3$:? #4,0:CLOSE #4:TR
AP 0
140 IF PEEK(73)=167 THEN 2750:? "ERROR
 #";PEEK(73):END
150 GRAPHICS 2:POSITION 1,1:? #6;"DO Y
OU WISH TO WORK ON DEDUCTIONS,":POSITI
ON 2,4:? #6;"SOURCES OF INCOME,"
160 DN$=STR$(PEEK(1790))
170 GRANDT=0:TB=0:TC=0:TD=0:TE=0:TG=0:
TI=0:TP=0:TT=0
180 POSITION 3,6:? #6;"OR QUIT?"
190 ? :? "Enter  D   S  or  Q ."
200 GET #3,B:IF B=83 OR B=115 THEN 214
0
210 IF B=81 OR B=113 THEN 2840
220 XIO 36,#1,0,0,D1$:XIO 36,#4,0,0,D3
$:OPEN #4,4,0,D3$:INPUT #4;ENTRIES:CLO
SE #4
230 GRAPHICS 1:? #6;"tax deductions, 1
9";DN$:POSITION 3,2:? #6;"choose a num
ber"
240 ? #6:? #6;"1ADD MORE ITEMS"
250 ? #6:? #6;"2DEDUCTIONS LISTED   ON
 THE SCREEN"
260 ? #6:? #6;"3DEDUCTIONS LISTED   ON
 THE PRINTER"
270 ? #6:? #6;"4TOTALS BY CATEGORY"
280 ? #6:? #6;"5CHANGING ENTRIES"
290 ? #6:? #6;"6SORT ROUTINES"
300 ? #6:? #6;"7FINISHED WITH       DE
DUCTIONS"
310 GET #3,B:B=B-48:IF B<0 OR B>7 THEN
 310
320 ON B GOTO 330,480,580,690,1060,136
0,2120
330 REM Data Input
340 GRAPHICS 2:? #6;"THE CATEGORIES AR
E:":? #6;"e=EDUC. EXP.":? #6;"p=PROF.
EXP.":? #6;"b=BUS. EXP."
350 ? #6;"g=DONATIONS":? #6;"i=INTERES
T PAID":? #6;"t=TAXES PAID"
360 ? #6;"c=CAPITOL INV.":? #6;"d=DEPR
ECIARED ITEMS":? #6;"x=FINISHED"
370 OPEN #1,9,0,D1$
380 ? "YOU WILL BE PROMPTED FOR EACH E
NTRY.  HIT A KEY":GET #3,A
390 ? "{CLEAR}CATEGORY";:INPUT CAT$:IF
 CAT$="X" THEN 470
400 ? "CHECK #";:INPUT CK$:IF CK$="R"
THEN CK$="BILL"
410 ? "PAID TO WHOM";:INPUT WHOM$
420 ? "FOR WHAT";:INPUT FOR$
430 ? "AMOUNT $";:INPUT A
440 ? "{CLEAR}THIS ENTRY IS ",CAT$,CK$
,WHOM$,FOR$,"$";A
450 ? "IS THIS CORRECT";:GET #3,B:IF B
=78 OR B=110 THEN 390
460 ? #1;CAT$:? #1;CK$:? #1;WHOM$:? #1
;FOR$:? #1;A:ENTRIES=ENTRIES+1:GOTO 39
0
470 CLOSE #1:GOTO 230
480 REM Deductions to the Screen
490 OPEN #1,4,0,D1$:GRAPHICS 0
500 FOR ITEM=1 TO ENTRIES
510 INPUT #1;CAT$:INPUT #1;CK$:INPUT #
1;WHOM$:INPUT #1;FOR$:INPUT #1;A:A$=ST
R$(A)
520 IF A=INT(A) THEN A$(LEN(A$)+1)=".0
0":GOTO 540
530 IF 10*A=INT(10*A) THEN A$(LEN(A$)+
1)="0"
540 ? :? CAT$,CK$,WHOM$,FOR$,A$
550 NEXT ITEM
560 ? :? "PRESS ANY KEY TO CONTINUE";:
GET #3,B
570 CLOSE #1:GOTO 230
580 REM Deductions to Printer
590 OPEN #1,4,0,D1$
600 TRAP 2830:M=1
610 LPRINT :LPRINT "                    FE
DERAL TAX DEDUCTIONS, 19";DN$:LPRINT
620 FOR ITEM=1 TO ENTRIES
630 INPUT #1;CAT$:INPUT #1;CK$:INPUT #
1;WHOM$:INPUT #1;FOR$:INPUT #1;A:A$=ST
R$(A)
640 IF A=INT(A) THEN A$(LEN(A$)+1)=".0
0":GOTO 540
650 IF 10*A=INT(10*A) THEN A$(LEN(A$)+
1)="0"
660 LPRINT CAT$,CK$,WHOM$,FOR$,A$
670 NEXT ITEM
```

```
680 CLOSE #1:GOTO 230
690 REM Totals by Category
700 OPEN #1,4,0,D1$
710 FOR ITEM=1 TO ENTRIES
720 INPUT #1;CAT$:INPUT #1;CK$:INPUT #
1;WHOM$:INPUT #1;FOR$:INPUT #1;A:N=N+1
730 IF CAT$="B" THEN TB=TB+A
740 IF CAT$="C" THEN TC=TC+A
750 IF CAT$="D" THEN TD=TD+A
760 IF CAT$="E" THEN TE=TE+A
770 IF CAT$="G" THEN TG=TG+A
780 IF CAT$="I" THEN TI=TI+A
790 IF CAT$="P" THEN TP=TP+A
800 IF CAT$="T" THEN TT=TT+A
810 NEXT ITEM
820 CLOSE #1
830 ? "Do you want the totals sent to
the Screen or Printer?":GET #3,B:IF B=
80 OR B=112 THEN 900
840 GRANDT=TD+TE+TG+TI+TP+TT
850 GRAPHICS 0:? "Total for category B
=$";TB:? "Total for category C=$";TC
860 ? "Total for category D=$";TD:? "T
otal for category E=$";TE:? "Total for
 category G=$";TG
870 ? "Total for category I=$";TI:? "T
otal for category P=$";TP:? "Total for
 category T=$";TT
880 ? :PRINT "The total for ";N;" dedu
ctions, not including C & B is: $";GRA
NDT
890 ? :? " HIT ANY KEY TO CONTINUE":GE
T #3,B:GOTO 230
900 REM Totals to Printer
910 TRAP 2830:M=3
920 LPRINT :LPRINT "          TOTALS, 19
";DN$:LPRINT
930 LPRINT "Total for category B=$";TB
940 LPRINT "Total for category C=$";TC
950 LPRINT "Total for category D=$";TD
960 LPRINT "Total for category E=$";TE
970 LPRINT "Total for category G=$";TG
980 LPRINT "Total for category I=$";TI
990 LPRINT "Total for category P=$";TP
1000 LPRINT "Total for category T=$";T
T
1010 REM Printing Grand Total
1020 LPRINT :LPRINT "                GRA
ND TOTAL, 19";DN$:LPRINT
1030 GRANDT=TD+TE+TG+TI+TP+TT
1040 LPRINT "The total for ";N;" deduc
tions, not including C & B is: ":LPRIN
T GRANDT
1050 GOTO 150
1060 REM Changing Deductions
1070 GOSUB 1330
1080 ? #6:? #6;" WHICH CATEGORY DO  YO
U WISH TO SORT ON (B,C,D,E,G,I,P,T)?":
INPUT CATCOMP$
1090 OPEN #1,4,0,D1$:OPEN #2,8,0,"D:SC
RATCH"
1100 FOR ITEM=1 TO ENTRIES
1110 INPUT #1;CAT$:INPUT #1;CK$:INPUT
#1;WHOM$:INPUT #1;FOR$:INPUT #1;A
1120 IF CATCOMP$<>CAT$ THEN 1150
1130 ? CAT$,CK$,WHOM$,FOR$,A
1140 GET #3,B:IF B=67 OR B=99 THEN 121
0
1150 ? #2;CAT$:? #2;CK$:? #2;WHOM$:? #
2;FOR$:? #2;A:NEXT ITEM
1160 CLOSE #1:CLOSE #2
1170 XIO 33,#1,0,0,D1$
1180 SCRATCH$(1,11)="D1:SCRATCH,":SCRA
TCH$(12,22)=D1$(3,13)
1190 XIO 32,#1,0,0,SCRATCH$:GOTO 230
1200 REM Making Changes to Deductions
1210 GRAPHICS 0:? "{CLEAR}";CAT$,CK$,W
HOM$,FOR$,A
1220 ? :? "Do you wish to:":? " 1 Del
ete the whole entry?"
1230 ? :? " 2 Change the Category?":?
 " 3 Change the Check #?":? " 4 Chan
ge to whom paid?":? " 5 Change the Fo
r?"
1240 ? " 6 Change the amount of the c
heck?":GET #3,B
1250 IF B=49 THEN GOSUB 1320:GOTO 1110
1260 IF B=50 THEN ? "What is the new C
ategory";:INPUT CAT$:GOSUB 1330
1270 IF B=51 THEN ? :? "What is the ne
w Check #?";:INPUT CK$:GOSUB 1330
1280 IF B=52 THEN ? :? "To whom was th
e item paid?";:INPUT WHOM$:GOSUB 1330
1290 IF B=53 THEN ? :? "For what was t
he item ?";:INPUT FOR$:GOSUB 1330
1300 IF B=54 THEN ? :? "How much was t
he item ?";:INPUT A:GOSUB 1330
1310 GOTO 1150
1320 ENTRIES=ENTRIES-1
1330 GRAPHICS 1:? #6;" YOU WILL BE SHO
WN  EACH ENTRY.":? #6;" "
1340 ? #6;"IF YOU WISH TO KEEP IT, TYP
E k;":? #6;" TO CHANGE IT, TYPE c.":RE
TURN
1350 REM Sort Routines
1360 OPEN #4,8,0,D3$:? #4,ENTRIES:CLOS
E #4:XIO 35,#4,0,0,D3$
1370 GRAPHICS 1:? #6;"  sort routines"
```

```
:POSITION 3,3:? #6;"choose a number"
1380 GRAPHICS 1:? #6;"  sort routines"
:POSITION 3,3:? #6;"choose a number"
1390 ? #6:? #6;"1 DEDUCTIONS SORTED
BY CATEGORY"
1400 ? #6:? #6;"2 DEDUCTIONS SORTED
BY CHECK #"
1410 ? #6:? #6;"3 DONE SORTING"
1420 GET #3,B:B=B-48:ON B GOTO 1430,19
20,2110
1430 REM Category Sort
1440 OPEN #1,4,0,D1$
1450 GOSUB 2860
1460 I=1:FOR J=1 TO ENTRIES
1470 TEMP$="

                    "
1480 INPUT #1;CAT$:INPUT #1;CK$:INPUT
#1;WHOM$:INPUT #1;FOR$:INPUT #1;A
1490 A$=STR$(A)
1500 TEMP$=CAT$:TEMP$(LEN(CAT$)+1)="
"
1510 TEMP$(LEN(TEMP$)+1)=CK$:TEMP$(LEN
(TEMP$)+1)="   "
1520 TEMP$(10)=WHOM$
1530 TEMP$(41)=FOR$
1540 IF A=INT(A) THEN A$(LEN(A$)+1)=".
00":GOTO 1560
1550 IF 10*A=INT(10*A) THEN A$(LEN(A$)
+1)="0"
1560 TEMP$(81-LEN(A$))=A$
1570 C$(I,I+SLEN1)=TEMP$
1580 I=I+SLEN
1590 NEXT J
1600 CLOSE #1
1610 REM Bubble Sort
1620 MAX=SLEN*(ENTRIES-1)+1
1630 FOR I=1 TO MAX STEP SLEN
1640 DONE=1
1650 FOR K=1 TO MAX-I-SLEN1 STEP SLEN
1660 KSLEN1=K+SLEN1:KSLEN=K+SLEN:KSLEN
SLEN1=KSLEN+SLEN1
1670 IF C$(K,KSLEN1)<=C$(KSLEN,KSLENSL
EN1) THEN 1700
1680 DONE=0
1690 TEMP$=C$(K,KSLEN1):C$(K,KSLEN1)=C
$(KSLEN,KSLENSLEN1):C$(KSLEN,KSLENSLEN
1)=TEMP$
1700 NEXT K
1710 IF DONE THEN 1730
1720 NEXT I
1730 GRAPHICS 0:POKE 559,SCR:? :? "{BE
LL}{BELL}{BELL}{BELL}HIT ANY KEY!!!!":
GET #3,B
1740 GRAPHICS 2:? #6;"DO YOU WANT THE
    SORTED LIST SENT TO THE sCREEN OR
THE    pRINTER?"
1750 GET #3,B:IF B=112 OR B=80 THEN 18
40
1760 IF B<>83 THEN 1740
1770 REM Sorted List to Screen
1780 I=1:POKE 82,0:POKE 83,40:GRAPHICS
 0
1790 FOR K=1 TO ENTRIES
1800 ? C$(I,I+SLEN1)
1810 I=I+SLEN
1820 NEXT K:POKE 82,2:POKE 83,39
1830 ? :? "END OF LIST; PUSH ANY KEY F
OR MENU":GET #3,B:GOTO 1370
1840 REM Sorted List to Printer
1850 TRAP 2830:M=2
1860 I=1:LPRINT "            FEDERAL TA
X DEDUCTIONS, 19";DN$:LPRINT
1870 FOR K=1 TO ENTRIES
1880 LPRINT C$(I,I+SLEN1)
1890 I=I+SLEN
1900 NEXT K:TRAP 0
1910 GOTO 1370
1920 REM Sort By Check #
1930 OPEN #1,4,0,D1$
1940 GOSUB 2860
1950 I=1:FOR J=1 TO ENTRIES
1960 TEMP$="

                    "
1970 INPUT #1;CAT$:INPUT #1;CK$:INPUT
#1;WHOM$:INPUT #1;FOR$:INPUT #1;A
1980 A$=STR$(A)
1990 TEMP$=CK$:TEMP$(LEN(CK$)+1)="   "
2000 TEMP$(LEN(TEMP$)+1)=CAT$:TEMP$(LE
N(TEMP$)+1)="   "
2010 TEMP$(10)=WHOM$
2020 TEMP$(41)=FOR$
2030 IF A=INT(A) THEN A$(LEN(A$)+1)=".
00":GOTO 1560
2040 IF 10*A=INT(10*A) THEN A$(LEN(A$)
+1)="0"
2050 TEMP$(81-LEN(A$))=A$
2060 C$(I,I+SLEN1)=TEMP$
2070 I=I+SLEN
2080 NEXT J
2090 CLOSE #1
2100 GOTO 1610
2110 CLR :GOSUB 2760:GOSUB 2790:GOTO 1
50
2120 REM Storing # of Entries
2130 OPEN #4,8,0,D3$:? #4,ENTRIES:CLOS
E #4:XIO 35,#4,0,0,D3$:GOTO 150
```

```
2140 REM SOURCES OF INCOME
2150 XIO 36,#2,0,0,D2$
2160 GRAPHICS 1:POSITION 1,0:? #6;"DO
YOU WANT TO...":? #6:? #6;"1. ENTER SO
URCES OF   INCOME"
2170 ? #6:? #6;"2. LIST SOURCES OF
INCOME":TINC=0:TFT=0:TST=0:TLT=0
2180 ? #6:? #6;"3. delete some
sources of income"
2190 ? #6:? #6;"4. done with sources"
2200 ? :? "Enter  1 ,  2 ,  3 , or  4
":GET #3,B:B=B-48:IF B<1 OR B>4 THEN 2
200:ON B GOTO 2220,2370,2500,150
2210 ON B GOTO 2220,2370,2500,150
2220 GRAPHICS 2:POSITION 2,1:? #6;"you
 will be       prompted for each   par
t of the entry."
2230 POSITION 0,6:? #6;"[ENTER 0 TO FI
NISH ]"
2240 OPEN #2,9,0,D2$
2250 ? :? "HIT ANY KEY TO CONTINUE":GE
T #3,B
2260 ? "{CLEAR}GROSS INCOME":INPUT INC
:IF INC=0 THEN 2350
2270 ? "{CLEAR}PAID BY WHO?":INPUT PAY
ER$
2280 ? "{CLEAR}PAID TO WHOM?":INPUT WH
O$
2290 ? "{CLEAR}FEDERAL TAXES PAID?":IN
PUT FT
2300 ? "{CLEAR}STATE TAXES PAID?  0 IF
 NO TAX PAID":INPUT ST
2310 ? "{CLEAR}LOCAL TAXES PAID?  0 IF
 NO TAX PAID":INPUT LT
2320 ? "This entry is $";INC,PAYER$,WH
O$,FT,ST,LT,"Correct (Y/N)?";:GET #3,B
:IF B=78 THEN 2260
2330 ? #2;INC:? #2;PAYER$:? #2;WHO$:?
#2;FT:? #2;ST:? #2;LT
2340 GOTO 2260
2350 CLOSE #2
2360 GRAPHICS 2:POSITION 1,1:? #6;"DO
YOU WANT TOTALS (Y/N)?":GET #3,B:IF B=
78 THEN 2160
2370 OPEN #2,4,0,D2$
2380 GRAPHICS 2:POSITION 1,4:? #6;"DO
YOU WANT THE    TOTALS ON THE SCREEN O
R THE PRINTER?":? :? "ENTER  S  OR  P
"
2390 GET #3,B
2400 IF B=80 THEN 2610
2410 REM Income Totals to Screen
2420 TRAP 2470:GRAPHICS 0
2430 INPUT #2;INC:INPUT #2;PAYER$:INPU
T #2;WHO$:INPUT #2;FT:INPUT #2;ST:INPU
T #2;LT
2440 ? INC,PAYER$,WHO$,FT,ST,LT
2450 TINC=TINC+INC:TFT=TFT+FT:TST=TST+
ST:TLT=TLT+LT
2460 GOTO 2430
2470 CLOSE #2
2480 ? "TOTAL INCOME=$";TINC:? "FEDERA
L TAXES PAID=$";TFT:? "STATE TAXES PAI
D=$";TST:? "CITY TAXES PAID=$";TLT
2490 ? :? "HIT ANY KEY TO CONTINUE":GE
T #3,B:GOTO 2160
2500 REM Deleting Entries
2510 GRAPHICS 1:? #6;" YOU WILL BE SHO
WN  EACH ENTRY.":? #6;"IF YOU WISH TO
KEEP IT, TYPE K"
2520 ? #6:? #6;"IF YOU WANT IT      DE
LETED, TYPE D."
2530 OPEN #2,4,0,D2$:OPEN #1,8,0,"D:SC
RATCH":TRAP 2570
2540 INPUT #2;INC:INPUT #2;PAYER$:INPU
T #2;WHO$:INPUT #2;FT:INPUT #2;ST:INPU
T #2;LT
2550 ? INC,PAYER$,WHO$,FT,ST,LT:GET #3
,B:IF B=100 OR B=68 THEN 2540
2560 ? #1;INC:? #1;PAYER$:? #1;WHO$:?
#1;FT:? #1;ST:? #1;LT:GOTO 2540
2570 TRAP 0:CLOSE #1:CLOSE #2
2580 XIO 33,#2,0,0,D2$
2590 SCRATCH$(1,11)="D1:SCRATCH,":SCRA
TCH$(12,22)=D2$(3,13)
2600 XIO 32,#1,0,0,SCRATCH$:GOTO 2160
2610 REM Income Totals to Printer
2620 TRAP 2830:M=4
2630 LPRINT "     Sources of Income, 1
9";DN$:LPRINT
2640 LPRINT "Gross     Paid by
            Paid to    Fed. Tax  Stat
e Tax City Tax   ":LPRINT
2650 TRAP 2720
2660 INPUT #2;INC:INPUT #2;PAYER$:INPU
T #2;WHO$:INPUT #2;FT:INPUT #2;ST:INPU
T #2;LT
2670 F$="

                "
2680 F$(1)=STR$(INC):F$(11)=PAYER$:F$(
40)=WHO$:F$(51)=STR$(FT):F$(61)=STR$(S
T):F$(71)=STR$(LT)
2690 LPRINT F$
2700 TINC=TINC+INC:TFT=TFT+FT:TST=TST+
ST:TLT=TLT+LT
2710 GOTO 2660
2720 CLOSE #2:TRAP 0:LPRINT
```

```
2730 LPRINT "TOTAL INCOME=$";TINC:LPRI
NT "FEDERAL TAXES PAID=$";TFT:LPRINT "
STATE TAXES PAID=$";TST
2740 LPRINT "LOCAL TAXES PAID=$";TLT
2750 ? :? "HIT ANY KEY TO CONTINUE":GE
T #3,B:GOTO 2160
2760 REM DIMENSION VARIABLES
2770 DIM D1$(13),D2$(13),D3$(13),SCRAT
CH$(22),DN$(2),F$(80),A$(10):SLEN=80:S
LEN1=79
2780 DIM CK$(4),CAT$(1),WHOM$(30),FOR$
(30),CATCOMP$(1),WHO$(10),PAYER$(30):R
ETURN
2790 DN$=STR$(PEEK(1790))
2800 D1$="D:DEDUCTNS.":D2$="D:TAXSPAID
.":D1$(12,13)=DN$:D2$(12,13)=DN$
2810 D3$="D:ENTRYTOT.":D3$(12,13)=DN$:
RETURN
2820 REM "PRINTER ERROR ROUTINE
2830 GRAPHICS 2:POSITION 0,1:? #6;"tur
n on your printer":? :? "HIT ANY KEY!"
:GET #3,B:ON M GOTO 230,1850,830,2380
2840 REM Finish of Program
2850 XIO 35,#2,0,0,D2$:XIO 35,#1,0,0,D
1$:CLOSE #3:GRAPHICS 0:END
2860 REM Screen Turn-Off
2870 GRAPHICS 2:? #6;"THE SCREEN WILL
TURNOFF,  please wait..."
2880 FOR T=1 TO 200:NEXT T
2890 SCR=PEEK(559):POKE 559,0
2900 IF SORTED THEN 2920
2910 DIM C$(SLEN*ENTRIES),TEMP$(SLEN)
2920 FOR I=1 TO SLEN*ENTRIES STEP SLEN
2930 C$(I,I)="

                    ":NEXT I
2940 SORTED=1:RETURN
2950 REM HINTS
2960 REM FOR RECIPTS, ENTER R FOR CHEC
K # & BILL WILL BE PUT IN FOR CHECK #
```



COPS AND BOMBERS

KEYSTONE KAPERS
by Garry Kitchen
&
KABOOM!
by Paul Willson
Activision

Reviewed by Ann McBain Ezzell

We justified buying our Atari 800 by saying that our son (then two and a half) would benefit from all the educational software. Well, the educational stuff comes and goes, but the big attraction for him remains the games. In fact, he is much better at most of them than we are, so I don't play against him very often.

Occasionally I encounter a game that even I can play. Keystone Kapers and Kaboom! are two such games. I am sure that there are expert players out there racking up high scores that would put mine to shame, but even a novice gamester can enjoy these Activision cartridges.

In Keystone Kapers, you control Keystone Kelly, who is in hot pursuit of Harry Hooligan, "notorious con-artist, thief and all-around-not-a-nice-person." Hooligan has escaped from Sing-Sing and is currently at large in Southwick's Emporium, where he has cracked the safe and made off with gold bags and currency-filled suitcases. In keeping with his unpleasant personality, Hooligan has a tendency to throw beachballs and toy biplanes at his pursuer. He may also set a shopping cart rolling, or use a cathedral radio as a blockade. Keystone Kelly must be ever-vigilant in his chase to avoid collisions. He only has a limited time to catch Hooligan, and colliding with objects will use up some of the available time. (In the case of the biplanes, it's "Farewell, Kelly") If Kelly and the three reserve cops all go "off duty", if time runs out, or if Hooligan escapes off the roof, the game ends.

You move Kelly through the Emporium using a joystick. With the red button, you can make Kelly duck under or jump over obstacles along

the way. (Don't jump over the loot - you get points for recovered loot if you manage to apprehend Hooligan.) There is a security scanner at the bottom of the screen which shows your position as a black dot and Hooligan's position as a white dot. It is important not to get ahead of Hooligan, because he can easily go back down to a lower floor, while you must wait for the elevator. Usually, by the time the elevator comes for you, Hooligan is long gone. You can go up to a higher floor in the elevator or on the escalators at the ends of the floors. If you go up to the roof before Hooligan gets there, you can kiss the game goodbye, because you cannot get off the roof and Hooligan will make his escape.

This game has 16 levels. At the higher levels, more and more objects come sailing at you at faster and faster speeds, and you must develop a feeling for their rhythms. You can pause and restart the game by pressing a key on the keyboard, but I seldom last long enough to need a break. Another nice feature is the ability to turn off the music which normally plays throughout the game.

The documentation for this game is rather unusual; it is in the form of the Blarney Beacon, a small town newspaper. A clever idea, but you do have to spend some time reading through it to find out all about the game. Instructions are included for Atari Home Computers, Atari 5200 and Colecovision.

This game is no Archon2, but it is playable and rather enjoyable. I doubt that it would interest hard-core game fanatics, but might be a nice choice for a family with children too young to handle the more complex arcade games and too old for Big Bird's Funhouse.

Kaboom!, on the other hand, is the sort of game that will find you saying "Just ONE more time - I know I can catch all those bombs!" An hour later, you will still be crouched in front of your computer, moving buckets of water back and forth in an attempt to thwart the Mad Bomber as he hurls bombs from the rooftop.

This deceptively simple game is addicting. It is well-engineered; the sort of game that takes a minute to learn and many hours (dare I say a lifetime?) to master. You start with three water buckets to control using a joystick or paddle controller. The OPTION key lets you choose both the type of controller and the width of the buckets. Pressing SELECT will let a single person play against the Mad Bomber, two players take turns against him, or two players play Pitch & Catch (alternating dropping and catching the bombs). These game variations allow you to practice in secret, building up expertise until you can vanquish your favorite opponent.

As you play, you may find yourself with a sudden craving for Quaker Puffed Rice. (Remember "the cereal that's shot from guns"?) As the bombs fall faster and faster, you will realize that the notes sounded each time you catch a bomb play the famous theme from "The 1812 Overture". The fireworks come in when you fail to catch a bomb; it detonates and sets off a chain reaction in the other bombs on the screen. Each missed bomb loses you a water bucket; for every 1000 points you score, you get another bucket, up to a maximum of three at one time. Each successive wave of bombs falls faster than the one before, but the waves do not start until you press the joystick button. The faster the bomb, the more points it is worth if you catch it.

As the bombs come hurtling down, it is often difficult to move the buckets fast enough to catch them if you are using a joystick. We don't have any paddle controllers, but discovered that we could use our Atari Touch Tablet, which reads the paddle ports, to control the buckets. Tell the computer that you want to use paddle controllers, and hold the Touch Tablet so that what is normally the bottom of the drawing surface is towards the computer (unless you can handle a left-for-right reversal and still catch bombs). You must be careful not to lift the stylus off of the Touch Tablet, or the buckets will slide off to the side of the screen. Once you get used to maintaining pressure on the Touch Tablet, I think you will find that you have better control than you do with the joystick.

Kaboom! is definitely one of those games that you will want to play over and over, and it's easy enough to learn to make it a good party game.

THE ATARI 800XL:
A Practical Guide

Reviewed by J. P. Sicola

I own the good ol' Atari 800 work horse, but with local retailers knocking the socks off the retail price of the 800XL ($88 at last look), I figure it's time to consider upgrading. With the kind of track record Jack Tramiel has had in previous computer companies, I didn't want to take a chance with an XE computer that might not be compatible. And by some accounts, the XLs won't be around much longer.

I sort of fell into receiving a copy of this book, and most assuredly will be returning it to its hapless owner. Copyrighted 1984 by Sybex Computer Books, the 198 page Guide was written by Thomas Blackadar. Mr. Blackadar, a Princeton graduate, has also written books for the Coleco Adam, 99/4A, & Vic-20. I'm glad I didn't purchase the book. Here are a couple of reasons why:

I feel I'm sort of experienced with my machine. This book takes you from Step One and assumes you are not computer-knowledgeable. That's fine for beginners, not me.

Much of the information about the XL series didn't come out until after the book was published. There was no mention of the translator disk, used to run pre-XL, non compatible software on the XL machines, although he mentions that some software is incompatible. The solution? Most "software companies have...prepared an updated version..." Sure they have.

The author mentions DOS 1.0 as if it were really out there. I haven't seen it in over five years of owning my 800/810. He also extols the joys of the (then) new Atari DOS 3.0, referring to it as having improvements over DOS 2.0, and stating that it is compatible with the old 810s (it ain't).

There are a lot of charts and pictures that could probably have been left out. Screen shots of PAC-MAN and other games aren't really necessary. A chart on page nine lists the differences between the 800XL and 1450XL. Lots of confidence, but little insight.

The middle section of the book tries to teach you BASIC programming techniques. Seems to be an all-right overview. Definitions and demonstrations of commands are given. Better books on programming can be found.

The book concludes with discussions of peripheral devices such as cassette and disk drives, printers, light pens, touch tablets, and, oh yes, the expansion box (that almost saw the light of day). No mention was made of third party hardware in this section or the book as a whole. A brief glossary of terms, the obligatory description of error codes and an index and Further Reading guide conclude the book.

The general tone of the book is upbeat. It is not written in the condescending manner we have come to know and despise - not intimidating in the least.

No price was given on the book (must have been a complimentary copy). I suspect it might be quite high ($5.00 -- $12.00 ???). As scant as some of the information that is provided with the Atari is, I don't see where this book could be of help. Your local dealer (even a K-Mart) could provide more information in most cases than this practically impractical guide.

(There is no mention of Atari's well-organized User Groups, either. These are probably the greatest source of help and education about your computer you can obtain.)

The rapidly advancing technologies we now experience have left this manual in their wake. Siskel and Ebert would not hesitate to give this dog "thumbs down". And neither will I.

Rating: Two woofs.

OPEN MINDED?

IMPATIENT FOR THE NEW ATARI PRODUCTS?
READY TO UPGRADE NOW?
TIRED OF BEING A "LONE RANGER"?
WORRIED ABOUT SOPHISTICATED NEW TECHNOLOGY, WITH NO SUPPORT?

LET US DEMONSTRATE SANYO 555

128 K 16-BIT DIRECTLY ADDRESSABLE (OPT. 256 K TOTAL)
360 K DOUBLE-SIDED, DOUBLE DENSITY DRIVE (OPT. 2ND DRIVE)
IBM STYLE KEYBOARD, ACCOUNTANT'S KEYPAD
80 CHARACTER HI-RES OUTPUT (OPT. COLOR)
MS-DOS OPERATING SYSTEM FOR HIGH IBM COMPATIBILITY
WORDSTAR (value $395) and CALCSTAR (value $250)
HI-RES SANYO MONOCHROME MONITOR
2 HOURS MS-DOS TRAINING
1 YEAR WARRANTY

SPECIAL MACE PRICE $995

(until May 21st)

****
* BUY ANY 3 PIECES OF ATARI SOFTWARE, GET ZAXXON FOR $4.95 *
* BUY ANY 5 PIECES OF ATARI SOFTWARE, GET ZAXXON FOR FREE!!! *
****

MANY OTHER NEW AND OLD TITLES AT FANTASTIC SAVINGS
SPECIAL ORDERS GLADLY TAKEN AT SPECIAL PRICES
AUTHORIZED ATARI SERVICE CENTER - FAST TURNAROUND FOR SERVICE
COMMODORE AUTHORIZED SERVICE FOR C-64'S AND VIC-20'S
WE HAVE SOFTWARE FOR IBM PC'S, COMMODORE C-64'S, VIC-20'S
ALL AT REDUCED PRICES FOR BUSINESS AND PERSONAL USE

FAMILY COMPUTER CENTER

YOUR ATARI CONNECTION
FOR ALL YOUR COMPUTER NEEDS
BERKLEY, MICHIGAN * (313) 543-0520

M-SS-NG L-NKS

HesWare

Reviewed by Michael Schiffer

D- y-- l-k- t- s-lv- p-zzl-s w-th m-ss-ng l-tt-rs? If so, M-ss-ng L-nks is the educational program for you. When the program is booted (it autoruns, but requires BASIC) you are given a series of questions on how you want to play: number of players (1-2), subject matter, book, and format. The subject matters include "Unreal worlds" (fantasy and sf), "Mysteries", "Limericks", "Humor" and others. There is then a choice of nine book excerpts, from such tomes as "Second Foundation" by Isaac Asimov, "The Old Man and the Sea" by Hemingway, and works by Tolkien, Twain, Huxley, as well as anonymous limericks.

Format indicates what letters you wish to be replaced by dashes (-). Choices include: vowels, every other letter, all but the first letter in a word, every other word, all consonants, all but the last letter per word, all but the first word per sentence, all words, or, finally, all letters with no dashes. Thus, difficulty spans from very easy to "photographic memory required". Subsequently you must choose the number of guesses (1-5) you would like before the computer fills in the letter for you. At this point the computer displays a screen full of deathless prose, sans whatever you chose to eliminate. After you have finished filling in letters (or failing to do so), the computer will give a percentage score for each player.

This program is one of the better packages for the person who reads Games magazine, or just the word games in the newspaper. While I am not certain I would class it as "educational", it is surely worthwhile for anyone who is "Pencilwise".

THE POND

HesWare

Reviewed by Michael Schiffer

The Pond is a pattern recognition game with elements of Frogger. Your mission is to guide your friend the frog across the pond by jumping on lily pads. You must do this by determining a two, three, or four step pattern from the pads on the screen, and hope it is correct for the entire path, which covers multiple screens. The game allows one or two players, 6 levels of difficulty, a practice mode, and built in instructions. While the box says "Ages 7 and up", the more difficult modes provide a challenge even to an ancient of 16. The program is very easy to use, but does require some reading skill. The instruction booklet is well written and includes a few pages on patterns and space for notes. Certainly this is a fine program to give the little munchkins when Frogger and Star Raiders start to pall and you want them to learn something.

MACE UNCLASSIFIEDS

[These ads are available free of charge to MACE members. Call in your ad to (313) 646-4455 two weeks before the meeting. Non-commercial ads only.]

FOR SALE

Atari 1027 Letter Quality Printer

Excellent condition. 1 year old, in original box. Used very little. Includes word processing software. Asking $190. Call Rich at (313) 545-8745 after 5pm weekdays, any time weekends.

BASIC Programming Class

To be held in Canton Twp. as soon as 12 people sign up and can agree on a schedule. 18 years or older only. Taught by MACE Journal Editor Ann McBain Ezzell. If interested, contact Jim Kronberg at (313) 981-2071.

GREETINGS FROM YOUR EDITOR...

Well, I'm still waiting for the submissions to come rolling in in response to last month's tirade. A few people interested in doing reviews have sent in samples of their work, but we could use a larger base from which to choose. If you would like to review products for the Journal and/or meetings, send a sample review and information about your system (specifically, what peripherals you use) and the types of products you feel qualified to review. Don't forget to include your name, address and telephone number.

I got another letter from a member who had some ideas for topics for the Journal. Some of these appear in this month's issue, and I will try to cover the others later. This member also had a couple of suggestions about the meetings. If you agree or disagree, let us know, and perhaps we can do something about it.

The first idea was to divide the monthly meeting into two groups: one for the novice and semi-beginner, the other for the experienced users and hackers. We have enough trouble filling up one agenda each month; I think we would need two Program Coordinators to handle two separate meetings. Maybe some interested new users could form a New User SIG.

We officers do realize that many of the members who come to meetings are relative newcomers to the world of Atari, and we would like to be able to help ease your way, but we do also have to try to interest the old-timers. How many times can you listen to an explanation of FOR-NEXT loops? One of the ways we have tried to please everyone is by having the question and answer sessions; we hope to be able to continue them. (Make a point of telling Dave Heinrich how much you appreciate his efforts.)

The second suggestion was that we cut down on the time dedicated to game demos at the meeting. I think that most people would agree that they would rather stay home and play Star Raiders than drive long distances at night to watch someone else play a game, no matter how interesting the game is. We will try to find other material to fill the meeting time - any volunteers? Program Coordinator Scott Garland is always ready for help.

One of the requests for information that came in this letter was for a list of the best books to buy for the Atari. I am a confirmed bibliophile, and some of my favorites might not appeal to everyone, but there are a few which are generally accepted as "classics". Several new books have come out recently, some worth the money and some that are real bow-wows (see the review by J. P. Sicola elsewhere in this issue). At any rate, here are my suggestions for stocking your computer bookshelf:

The very first book you should buy, even before you crack the seal on your Atari box, is Your Atari Computer by Lon Poole (Osborne/McGraw-Hill). The newest edition includes information about the XL series machines. This book is full of information about the Atari, including the program recorder, disk drive, and Atari printers. There are two chapters on BASIC programming, two covering graphics (including player/missile) and one on sound. Many sample programs illustrate the material in the text. The last chapter is a comprehensive summary of BASIC statements and functions with lots of charts and explanations. There are also nine appendices covering error messages; codes, characters and keystrokes; memory usage; useful PEEK and POKE locations; and more. If you can only afford one book about your Atari, buy this one. It's well worth the $17 (more or less).

I have two books which could serve as teaching guides for Atari BASIC. Back when I bought my 800, Atari was still packing Atari BASIC by Bob Albrecht, LeRoy Finkel and Jerald R. Brown (John Wiley & Sons) with the BASIC Reference Manual and the BASIC cartridge. I started learning BASIC from this book, and it gave a reasonable foundation, but it is not terribly Atari-specific (doesn't even mention player-missile graphics, for example). I don't know if it is still in print, but if you could pick up a used copy for $5 or less, and are looking for something to get you started, you might want to make the small investment.

A better choice would be The Elementary Atari by William B. Sanders (Datamost). This is the book that sometimes comes with Indus GT disk drives, but it is also available in stores or from Datamost. This spiral-bound guide will take you through hooking up your Atari, writing programs, and dealing with cassettes, disks and printers. The text is studded with offbeat line drawings and helpful hints in boxes. The one thing that seems to be missing is a list of error messages. The text refers to Appendix A, which is supposed to contain such a list, but my copy had no Appendix A. In spite of this omission, I think that this is another winner for your bookshelf. You could learn a lot about programming in BASIC, and pick up many hints about using your Atari.

Another Datamost publication is the ABCs of Atari Computers by David E. Mentley. This book has been offered as a subscription premium by ANALOG Computing and is also available from ANTIC Magazine. If you are lucky, you will walk into your local computer store just as they are opening the carton and snatch one up before they disappear. There is a review in this issue of the Journal, but let me say again here that this is one of those books you should chain to your computer desk.

While I am thinking about ANALOG, I should mention their Pocket Reference Card. This was included in subscription copies of a recent issue and is also available from ANALOG for $7.95. Too bad if you missed the free copy, but you should really consider laying out the money for a copy of your own. It certainly beats any of the other reference cards I have seen. It covers BASIC instructions, XIO command codes, joystick movement, sound commands, player/missile layout, symbol device names, 6502 assembler language mnemonics, frequently used PEEK/POKE locations, error messages, color register values, ATASCII code translations, internal codes (PEEK(764)), hex/decimal conversions, default character sets, and graphics mode specifications - and it folds up tiny!

Now we come to THE Atari sourcebook! Ian Chadwick's Mapping the Atari (Compute! Books). As I have said, no one who programs the Atari should be without this book. It is a thoroughly annotated memory map, full of more information than you will ever need to know about the Atari. (It even has an Introduction by the venerable Bill Wilkinson, in which he writes and discusses a short programming example in seven Atari computer languages.) I can't think of anything to tell you about this book that would convey how useful it is; buy one and you'll quickly find out for yourself.

If you are ready to tackle more complex programming topics like display list interrupts, scrolling and advanced sound generation, you should try to find a copy of De Re Atari by Atari's Software Development Support Group. This manual was distributed by the now-defunct APX (Atari Program Exchange); I don't know if it's still available. You might be able to borrow one. Be prepared for some intense reading, though; this is NOT a book for beginners. The Preface identifies the book as a "training manual for professional programmers who use the Atari Home Computer" - believe it. I was a novice when I first got De Re, and I read and reread it many times over before some things started to sink in. Still, there are jewels in this book worth digging out.

Also on the heavy side are the Technical Reference Notes, consisting of the O.S. User's Manual and the Hardware Manual. I doubt that any but the most serious hackers would get any use out of these ponderous "notes".

Compute! Books have several books on the Atari in addition to Chadwick's memory guide. There are at least four which are primarily compilations of articles from the Compute! magazine, imaginatively entitled Compute!'s First Book of Atari, Second Book of Atari, First Book of Atari Graphics, and (you guessed it) Second Book of Atari Graphics. I believe there is also a book about writing games. If you don't have the back issues of Compute! (and even if you do, but are too lazy to dig through them), you can learn quite a bit from these books. Compute! also has published a couple of books about machine language programming.

And then there are the books about different languages... but you have probably blown your book budget for this month by now. See you next month.

SAVE UP

TO 50%

SECTOR

INTERNATIONAL

ONE

MMG
ICD
BRODERBUND

LEGEND

SEND FOR
FREE CATALOG

AND RELEASE
THE MAGIC IN
YOUR COMPUTER

ATARI

GEMINI

COMPUTER MAIL ORDER *WORLDWIDE*

37220 TRICIA DRIVE, STERLING HEIGHTS, MI. 48077

ORDER INFO. Phone: 313-978-2208 9am - 7pm. EST

SECTOR ONE New Products for *ATARI*

130XE COMPUTER ♥ NEW $179.95	ATARILAB STARTER ♥ NEW $ 49.40	ATARILAB LIGHT ♥ NEW $ 32.50	MILLEPEDE CART ♥ NEW $ 17.25
MOON PATROL CART ♥ NEW $ 16.88	SKYWRITER CART ♥ NEW $ 16.99	TRACK and FIELD♥ ♥ NEW $ 21.92	♥ PRINT SHOP ♥ ♥ NEW $ 39.95
KOALA PAD A or C ONLY $ 49.95	US DOUBLER K ROM 1050 DD $ 64.95	TRACK BALL LOW $ 22.94	850 INTERFACE PRICE $119.94
VISI CALC ↓ ONLY $ 49.99	→ SYN CALC ← JUST $ 39.96	→ SYN FILE+← LOW $ 39.96	→ SYN TREND← PRICE $ 39.96
600XL COMPUTER SALE $ 39.95	800XL COMPUTER SALE $ 99.95	1200XL COMPUTER* SALE $ 79.95	1020 PRINT/PLOT* SALE $ 49.95
1010 RECORDER ONLY $ 39.94	1025 PRINTER JUST $189.94	1027 PRINTER SALE $249.94	1030 MODEM W/SW. PRICE $ 74.84
TOUCH TABLET ONLY $ 44.94	CX85 KEY PAD JUST $ 29.94	1050 DISK DRIVE LOW $179.94	I/O CABLE 6ft. PRICE $ 12.95
HAYES S.M. 300 ONLY $189.94	ATARIWRITER♥ SALE $ 35.73	MMG COMPILER LOW $ 74.80	15# CASE PAPER →3300← $ 24.95

DON'T SEE IT? PLEASE CALL - 313-978-2208

MICRO LEAGUE BASEBALL

Micro League Sports Association

Reviewed by Scott Menig

Micro League Baseball is a one or two player game. The list price is $39.95, but it can be purchased at most computer stores for around $30.00.

This is a unique baseball game, because instead of playing it, you manage it. Micro League Baseball instructions are self-explanatory, and should have you winning games in no time. The game itself contains brilliant graphics. The baseball field is a perfect replica of a real baseball field. The players are also very realistic in their own way. One of the nice features of this game is that you don't have to worry about controlling the players. All you have to do is hit a few keys to pick your play, and you're done. The computer will randomly give your batter a count, like: 2 balls, one strike. This feature makes the game quick to play and more enjoyable.

Besides the great graphics, Micro League Baseball also has great sound effects, like the roar of the crowd and the crack of the bat. It even plays the national anthem at the start of the game, and has a seventh inning stretch.

Micro League Baseball comes programmed for 26 baseball teams from the past - teams like the '68 Tigers, the '61 Yankees and the '75 Reds. More teams can be added for additional fees. You chose a team you'd like to manage, then you can pick the line-up, chose a starting pitcher, make pitching changes and much, much more for that team.

If you're one of those people who sit at a ball game and second guess the managers, this game is perfect for you. I personally feel that this is a very good game.

ABCs OF ATARI COMPUTERS

by David E. Mentley

Datamost

Reviewed by Ann McBain Ezzell

Imagine my joy when I walked in to the Family Computer Center the other day and saw the ABCs of Atari Computers - I had seen this new book mentioned in ANALOG Computing, and it seemed like a good addition to my ever-growing computer library. Regardless of the size of your library, I would strongly recommend that you buy this book. The front cover claims it is "The definitive reference source for owners of Atari Home Computers", which is as close to the truth as anything you are likely to read on a book's cover.

ABCs is organized, appropriately enough, alphabetically by subject - from "A" for accumulator to the "Z80" microprocessor, with lots and lots in between. Somewhere in the middle is even an entry for MACE, complete with correct address and a plug for the Journal. The entries cover elementary topics like "READY" and "SAVE", but also delve into such intricacies as a hardware modification for the 810 disk drive which overrides the write protect feature and lets you write to the back side of a disk without notching it.

As far as I could tell, most of the information given is accurate, but I did notice a few items which were not exactly true and which might mislead a beginner. The "CAPS LOWR DISABLE" entry, for example, claims that a POKE 702,64 statement will prevent accidentally switching to lower case. That POKE will, in fact, return keyboard input to uppercase, but only until such time as the CAPS LOWR key is again pressed. To filter out lower case input, you either need to write a routine to change lower to upper case or you need to monitor location 702 and POKE in 64 when necessary.

In spite of the few misleading items, and the fact that the book should really be spiral bound to lie flat on your desktop, this is definitely a "Must Buy" for anyone who owns an Atari computer.

PAGE ZERO

by Ann McBain Ezzell

A common complaint that new (and old) Atari users have is that the error messages are simply numbers, with an occasional reference to a line number. You write a program, run it, and suddenly you are faced with an "ERROR 3 AT LINE 210". XL owners can pull out their Atari BASIC Reference Guide and look on page 13 to discover that ERROR 3 means "Value Error". Big help. What does that mean? You can get more information from Lon Poole's Your Atari Computer (Osborne/McGraw-Hill); it will tell you that "A numeric value is too large, too small, or negative when it should be positive". ABCs of Atari Computers by David E. Mentley (Datamost) will further enlighten you by stating that ERROR 3 is a "Value Error such as a result outside the range of +/-10 to the 97th power (or to the 127th power with the Newell Fastchip)". With all this edification, you still may be left wondering just what the problem is, and how to fix it.

This month's column will be a discussion of some of the Atari error messages with suggestions for debugging. We will cover messages 2-21, which relate to problems with a BASIC program. Errors 128-171 deal with input/output problems and may be covered at a later date.

There are two memory locations which are important in dealing with errors: 195 (ERRSAVE), which contains the error code number generated, and 186,187 (STOPLN) which together hold the line number at which the program stopped due to an error, the BREAK key, or STOP or TRAP statement. These locations can be used when you write your own error-trapping routines, and can be useful when debugging. They can be accessed either from within your program or from immediate mode as follows:

```
10 ERRNUM=PEEK(195)
20 LINE=PEEK(186) + 256*PEEK(187)
30 PRINT "ERROR NUMBER ";ERRNUM;
40 PRINT " AT LINE ";LINE
```

Now for the codes themselves:

ERROR 2 - Insufficient memory
You don't have enough space in RAM to store or run your program. This might occur while typing in a long program (especially if you only have 16K RAM), or if you try to dimension a large string or array, or if you try to use a graphics mode that requires a lot of memory (Graphics 8 and above, for example). You can use the PRINT FRE(0) statement to see how much space is available to you. You may have enough room to load in a program, but will get ERROR 2 when you try to run it because the program dimensions a large variable or uses a memory-eating graphics mode.

If you only have 16K, consider expanding your RAM. You will probably want to do this at some point, anyway - why not now? Many commercial programs require more than 16K, especially those on disk.

If you are writing your own program, try this: LIST the program to tape or disk, type "NEW", ENTER the program back into memory, then SAVE (or CSAVE) it again. This will clear any unused variables out of the Variable Name Table, which is included when you SAVE a program. LISTing only includes the program code itself. If you have made a lot of changes while writing your program, you may find quite a savings in memory by doing this. (Check with FRE(0) before and after to see how much you have saved.)

You can save memory in your program by using shorter variable names and by putting more than one statement on a line. (Put colons between the statements to separate them.) Using strings to store data (as ATASCII characters) is more efficient than using arrays. If you use a number more than three times, make it a variable (actually a constant). Instead of 0, 1, and 2, use Q0, Q1 and Q2. Your code will be clear and you will have more room.

ERROR 3 - Value Error
A number was too big or too small, or negative when it should have been positive. This occurs if you try to POKE a negative value or a value greater than 255 into a memory location. It is most likely to happen when you are using variables in a POKE statement and something unexpected happens to the variable.

Look at the program line where the error occurred. PRINT out all the variables referenced in that line; chances are that one of them is too big or negative. You may have to look elsewhere in the program to determine where the variable took on its bad value. Check lines where variables are manipulated (A-B, C/D, etc.) You might want to insert PRINT statements in the program at critical locations to check the variable values.

ERROR 4 - Too many Variables
BASIC allows a maximum of 128 variable names. Try the LIST/NEW/ENTER method given above to remove unused names. If that doesn't work, try to combine related variables into an array. For example, if you are storing ten bowling scores as SCORE1, SCORE2, SCORE3, etc., dimension an array SCORE(9). You can then reference the elements in the array by calling SCORE(I), where I ranges from 0 to 9. You might find that this method will let you write a cleaner and faster-running program, also.

ERROR 5 - String Length Error
You have tried to read or write beyond the dimensioned length of a string. Remember that strings start with 1 as an index, as opposed to arrays, which start with 0. This error is most likely to occur when you are referencing a string with variables (for example, filling a larger string with substrings in a loop). Check the values of your variables and compare them to the dimensioned length of the string. If necessary, change the string dimension.

ERROR 6 - Out of Data Error
You do not have enough data for your READ statement(s). You probably miscounted the number of data items, or left out a comma between two items. When reading the same data more than once, be sure to use RESTORE to set the pointer back to the first DATA statement to be read.

ERROR 7 - Line Number Greater Than 32767
Unless you simply mistyped a line number, you probably have referenced a line using a variable (GOTO TEST, GOSUB GETKEY or RESTORE FIRSTDATA) with a value greater than 32767. Print out the values of variables used in this way.

ERROR 8 - Input Statement Error
When BASIC is expecting a numeric input (INPUT X, as opposed to INPUT X$), you cannot include any letters, punctuation, or graphics characters in the input. This error will also occur when you simply hit <RETURN> in response to a numeric input request.

Unless you are writing a short program for your own use and you know that you will always be entering numbers, it is best to put some error-trapping routines into programs which expect numeric input. One way to do this is to use the TRAP statement to send the program to a routine which checks for errors whenever they occur. You will probably want to use PEEK(195) to determine the error code and base the reaction on the code. Such a routine might print a message like "NUMBERS ONLY, PLEASE" and then return to the input routine. A quick TRAP is to send the program back to the same line:

```
10 TRAP 10: INPUT A
```

This method is not particularly elegant, since it does nothing to explain to the user why it won't accept the input.

You could also accept your input in string form, loop through the string to check for non-numeric characters, and use the VAL function to turn the string contents into a number. This method is rather cumbersome, but it could be done.

ERROR 9 - Array or String DIM Error
You will probably encounter this most often when you try to reference an undimensioned string or array, either because you added a variable and forgot to dimension it, or because you mis-typed the variable name. It's a good idea to put all of your DIM statements together either at the beginning of your program or in an initialization subroutine, so that you can check them easily.

This error will also result if you attempt to dimension an array larger than 5460 or a string longer than 32767, or if you try to redimension a variable without using the CLR statement. (Be careful with CLR; it will undimension all strings and arrays and wipe out the values in all of your variables.)

ERROR 11 - Floating Point Overflow/Underflow
You have attempted to divide by 0 or to refer to a number of magnitude 1 E+98. This is most likely to occur when dividing by a variable which is unexpectedly equal to zero. Check variable values using the PRINT statement.

ERROR 12 - Line Not Found
A GOSUB, GOTO or THEN statement tried to go to a non-existent program line. If you are using a line number rather than a variable after the GOSUB, etc., you probably inadvertantly deleted the target program line. Check the most recently saved copy of your program. When using variables with GOSUBs, etc., you might not have the value in the variable that you expect to have.

ERROR 13 - No Matching FOR
A NEXT statement was encountered without a matching FOR. Be careful when nesting FOR-NEXT loops not to cross the loops. The following lines show proper loop nesting:

```
10 FOR I=1 TO 10
20 FOR J=5 TO 50
30 PRINT I*J
40 NEXT J
50 NEXT I
```

The first FOR statement is matched with the last NEXT statement, so that each successive loop is contained within the previous one.

If you use POP without a preceding GOSUB within a FOR-NEXT loop, you will disable the most recently executed FOR statement. Also watch out if you must jump around within nested loops that you do not enter a loop after the FOR statement.

ERROR 14 - Line Too Long Error
The program line is too complex or too long for BASIC to handle. If you encounter this error, break the line up into two or more lines. (In all my years of causing distress to my Atari, I have never come across this error message.)

ERROR 15 -GOSUB or FOR Line Deleted
Lost in the funhouse - a NEXT or RETURN statement can no longer find its corresponding FOR or GOSUB command. This is another which I have not encountered. I suppose it could happen if you wrote a program which modified itself by deleting line numbers during program execution. Some programs do this to allow more room for special graphics modes, player-missile graphics, redefined character sets, or buffers. If you delete program lines while the program is running, be careful.

ERROR 16 - RETURN Error
The program encountered a RETURN with no GOSUB. This is similar to ERROR 13. Put an END or GOTO statement before your subroutines so that the program will not accidentally fall through into the subroutines and find an orphan RETURN.

ERROR 17 - Syntax Error
This is also less respectfully known as the "Garbage Error". A program statement has for some reason become garbled and meaningless. A misdirected POKE statement or a machine language routine might have interfered with the program as stored in RAM. If you are typing in a program and make a mistake while typing, the line will be printed to the sreen with an error message. Unless you retype the line, it will be stored with the error message included and will cause ERROR 17 when you try to run the program. Be sure to retype the line and hit RETURN to enter it into the program.

ERROR 18 - VAL Function Error
The program tried to convert a non-numeric string to a numeric value. Check your program line carefully for typos.

ERROR 19 - LOAD Program Too Long
Your program is too large to fit into the available RAM. See suggestions under ERROR 2 to shorten the program. Also, some versions of DOS take up less RAM than others; you might need to switch to a different DOS.

ERROR 20 - Device Number Error
The program tried to use channel 0 or a channel greater than 7. Check the values of any variables used to reference channels.

ERROR 21 - LOAD File Error
You tried to load a nonload file. Only files which have been stored using the SAVE command can be LOADED. Use CLOAD with CSAVE and ENTER with LIST. Binary files can be entered into RAM from DOS with option L.

If you get this error, you can try the other loading commands or you can look at the file itself and determine what type of file it is. One way is to go to DOS and use 'C' to copy the file to the editor (type "D:FILENAME,E:", without the quotes, after C).

LISTed files will appear just as they do when listed from BASIC. SAVEd files will start with the variable name table, which will be a string of uppercase characters, some of which will be in inverse video. Most of the rest of the program will be a jumble of graphics characters. You may be able to read some REMarks.

The first two byte of binary load files are 255 (type ESC CTL-INSERT to see this character, which is an inverse video right-pointing triangle). Since binary files are likely to be full of graphics control characters that will send the cursor flying all over the screen, it may be hard for you to look at this type of file. I find that hitting CTL-1 to freeze the display immediately after the file starts printing will usually let me examine the first two bytes.

Whenever possible, it is a good idea to use extenders which indicate the file type. Some commonly used extenders are:

.BAS for SAVEd BASIC files
.LIS or .LST for LISTed files
.BIN or .OBJ for binary files
.SRC for source code
.TXT or .DOC for text files

This can prevent some confusion when you come back to a program after not using it for a while.

I hope that these explanations have been useful. As you get more experienced in programming, you will find that you will get fewer error messages, and you will more easily be able to find and correct them when they do occur.

This column was the result of a suggestion by a MACE member. If anyone else out there has any ideas for future columns, please send them in. You may not be the only one who wants that topic covered.

SO YOU WANT TO SELL YOUR PROGRAM?

[The following questions are from a document entitled GUIDELINES FOR SUBMITTING SOFTWARE TO ELECTRONIC ARTS and may help you to decide if your latest program is ready for submission, or needs more work. The folks at Electronic Arts are known for the excellence of their software; if you can meet their standards, you could be well on your way to success. -Ed.]

There are now over 5000 software development companies in the United States alone, and some days we feel like every one of them has a submission sitting in our mail room. In that environment, you want everything a publisher sees from you to be of the highest possible quality. That means asking yourself these questions:

* Is this product pretty good? Very good? One of the best 100 products on the market? If your answer to any of these questions is "yes", don't send it yet. The personal computer software market is so crowded that the only titles that have any real chance of breaking in are those that can be considered awesome, earth-shaking, world-class. You don't want a publisher to see your "ok" version, then the next "pretty good" version, then the next, and so on. Wait until you really feel it's awesome: that way you make the good impression the first time we "meet" you. This doesn't necessarily mean your product has to be finished; it just has to be complete enough to suggest its full potential.

* Is this my best idea? Many times Artists work on a project for months, struggling to make it something special, and think of other ideas which are set aside "until this one's done." All too often, that means that an OK project keeps you from ever getting down to work on one that could be a world-mover. We don't want to see the one that's "in the way"; when it's really ready, we want to see the one that's the best.

* Is this product unique? With the hundreds of submissions we receive, you can imagine how

many times we see virtually the same product go by. The world is not clamoring for another graphics/text adventure game, climbing game, maze game and any number of other genres. To get the market's attention -- and ours -- you have to be different.

* Is this concept something I can communicate on paper or will I have to program it to communicate it? We see lots of ideas like, "This game will simulate the whole world," and "This game is an exciting action product that features state-of-the-art graphics." Both of those statements will be met with skepticism, since saying it is one thing, doing it quite another. In most cases, it's much better to show people what you can do rather than tell them what you can do.

* What does the computer bring to the party? We see lots of books-on-computer, card-games-on-computer, board-games-on-computer, etc. The novelty value of computerizing something that worked fine without one has long since worn off, and we don't want to see such products. Conversely, something that drove people crazy until a computer made it convenient is a product with great potential.

* Does Electronic Arts publish this kind of product? Although rules are made to be broken, we aren't particularly interested in: software aimed primarily at children under 10 years old; software that teaches by drill and repetition; text-only entertainment or education software; software written in BASIC. As with all publishing media, the best way to find out what a publisher buys is to look at what they've already bought: products like Pinball Construction Set, Archon, Seven Cities of Gold, Dr. J and Larry Bird Go One-on-One and Music Construction Set. Look for the common elements in those products and you'll find the key to selling to Electronic Arts.

* Will anyone buy this a year or eighteen months from now? That's how long it takes to get the typical product from concept to marketplace on a number of machines. If you're trying to ride the wave of excitement over the latest fad, it's already too late to start.

* Will this product run on a number of computers? With so many different brands of computers in peoples' homes, no one machine can make a title really successful. If your product uses special features that are available only on your favorite home computer and which can't be simulated by other machines, its chances for commercial success are slim.

* Does this product need special hardware? Except for a disk drive and a joystick, there are few standard peripherals for home computers. Those programs that need the Fribish Technologies Quantum Physics Board and Nightlight are fighting an uphill battle. Even such widely-owned items at the Mockingboard, Koala Pad, etc. should be supported only as options.

* Do I know the marketplace? If you haven't published software before, how much research have you done? There is a sad truth in the industry right now that perhaps 1% (yes, that's one per cent) of the software that's published -- let alone created -- is actually selling well. In a crowded marketplace, writing software before you research your market is like trying to walk from New York to San Francisco on a nice day; the immensity of completing the task isn't changed by the fact that starting it is easy and feels like fun.

* Am I capable of creating this product? If you can't program on the machine that the first version of the product will run on, don't send us the idea until you have a programmer working with you who can. If you have to learn assembly before creating the product, wait until you've mastered it before you contact us.

* A final word from our Artists: when asked to give advice on how to succeed in this business, industry leaders like Bill Budge, Dan Bunten, Eric Hammond, Jon Freeman and Anne Westfall always tell people something like this:

"Be your own worst critic. The marketplace is cruel, hard and incredibly demanding. The only way your work will stand up is if you're cruel, hard and incredibly demanding of yourself before you get there."

PLANETFALL

Infocom

Reviewed by Jim Kennedy

Planetfall is a text adventure game by Infocom. It is available on disk only and requires 48K RAM. Infocom makes 14 other text adventures for the Atari home computers at present. They are: ZORK I, II, & III, Enchanter, Sorcerer, The Witness, Suspect, Deadline, Seastalker, Cutthroats, Infidel, Starcross, Suspended and The Hitchhiker's Guide to the Galaxy. Planetfall is one of the newer adventures and is a science-fiction/comedy.

What is a text adventure game? It is a game with no graphics. You must use your imagination to see what is going on. In the adventure, you will have a role to play. Your character will follow instructions which you type into the computer. The instructions may resemble: PUT CANTEEN UNDER SPOUT, DRINK LIQUID or E (short for GO EAST). You are the character as far as the computer is concerned, but must relay what you want to do through the keyboard. You can type in whole sentences and Planetfall will most likly understand what you want to do. If it doesn't, it will print something like: I DON'T UNDERSTAND THAT SENTENCE. Then you may have to rephrase and try again.

Probably the best feature of an Infocom adventure is the SAVE/RESTORE command. If you are about to try something dangerous, you can SAVE the game at that point. That way if you get killed (which can happen often), you can RESTORE it where you left off. There are also specific commands which allow you to get a printout of any text that scrolls by, list what you are carrying (INVENTORY), examine your surroundings (LOOK) and check the health of your character (DIAGNOSE).

In Planetfall your objective is to save a planet! But who are you to save it? Well, you're actually an Ensign 7th class (the lowest). At the start of the game you are on the Starship Feinstein, but not for long - your ship explodes! (Fortunately, not before you get off.) Once you escape to the planet below you must explore a sprawling complex. You will find many different items. As you roam around you will find puzzles that need to be solved, such as a locked door, malfunctioning computer or a fused 90 ohm bedister. These puzzles may be solved using the items you find, which include a piece of paper, a ladder and a can of spam. Don't expect every item you find to be useful. And some puzzles may never be solved.

Planetfall may take you weeks or months to complete, depending on whether you're an experienced adventurer or a beginner. You will probably die many times. Be patient and try anything that comes to mind, even if it doesn't make much sense, because you may learn something. Two very important hints: examine everything and make a map (using graph paper is best). You can't carry everything so be selective.

This was my first text adventure game, and I thought it was quite enjoyable. With the descriptions that were given, I found that I had no problem in visualizing what was going on. With me, the game had staying power. Even after I left the keyboard, I found myself trying to think up possible solutions to unsolved problems. It really made me feel like I was there. As my character, I had to eat, sleep and find companionship. (Which I did when I found Floyd, a comical robot. Although I couldn't get much out of Floyd in the line of dialogue, he did serve an important purpose in the adventure. As problems had solutions, he was a solution. But in the process, I lost a good friend.)

I only found one drawback to the game. When I entered commands and it didn't understand what I wanted, I had to keep trying different ways of saying it. Sometimes it still didn't understand what I wanted and at times that was aggravating. (Some people think this is a part of the fun.) However, this isn't something that would keep me from purchasing another of the Infocom adventures.

MC68000
MOTOROLA'S
16/32 BIT MICROPROCESSOR

by SIG-ATARI's Tim Barr

[Attention all you hackers out there! Are you anxious to get at the new ST Line of Atari Computers? Here's a little background info about the heart and soul of these new computers. This article was taken from the SIG Atari section of CompuServe.]

The MC68000 is a 16-bit MPU with 17 general purpose 32-bit registers, a 32-bit program counter and a 16-bit status register. The first eight GP registers (D0-D7) are used as data registers. The next seven GP registers (A0-A6) and the system stack pointers can be utilized as address registers and pointers for software use. The data registers can be used for different data sizes. These sizes are: BYTE (8-bit), WORD (16-bit), and LONG WORD (32-bit) operations. The MPU has a 24-bit address bus (actually it is 23-bit address line and a HI BYTE and LOW BYTE select lines which gives you 8 megawords or 16 megabytes of memory) and a 16-bit data bus.

Five basic data types are supported by the instruction set. These data types are: BITS, BCD DIGITS (4-bits), BYTES (8-bits), WORDS (16-bits), and LONG WORDS (32-bits). The MPU has 14 address modes of six basic types: REGISTER DIRECT, REGISTER INDIRECT, ABSOLUTE, PROGRAM COUNTER RELATIVE, IMMEDIATE, and IMPLIED. The surprising thing about the 68000 is that it only has 56 instruction types and a total of only 88 actual instructions. The actual 16-bit OP-CODE that the system uses is a combination of an instruction and an addressing mode, GP register number, an OP-MODE, instruction specific data, or any combination of the four. (Instruction specific data is such information as shift direction, branch conditions, operation size, etc.) This provides you with over 1000 acutal instructions, but keeps the total number of instructions small.

I will try to upload a list of the basic instructions of the 68000 [to CompuServe], but I wanted to mention a few of them here. The MPU can perform add and subtract functions on BCD digits in groups of two digits. It can also multiply and divide binary numbers in these formats: 32-bit / 16-bit => 32-bit and 16-bit * 16-bit => 32-bit. The signed divide is the slowest instruction to execute on the 68000. It takes 170 clock cycles or 21.25 microsec. at a clock speed of 8 MHZ. This means that you could divide any 32-bit number in memory by any 16-bit number in memory over 47,000 times in ONE SECOND!!

I hope that this has given you a general idea of the power of the processor that the new Ataris will be using. If you are interested in more information, I would like to suggest a couple of books that are available. The first is published by Howard K. Sams & Co. and is called "68000: PRINCIPLES AND PROGRAMMING" by Leo Scanion. You should be able to order this book through the Howard Sams section of CompuServe (GO SAM). The second book I would like to recommend is "M68000 PROGRAMMERS REFERENCE MANUAL" available from:

Motorola Semiconductor Products Inc.
P.O. Box 20912
Phoenix, AZ 85036

Ask for document # M68000UM(AD4).

ATARIMUSIC SIG NOTES

by Mike Lechkun
SIG Chairperson

The AtariMusic SIG has not met with great popularity as of yet. Only three members are now active. Over the past four months we have assembled to play some music for each other. We wish to change that for the future.

We'd like you to join. We work primarily with Advanced Music System, but also work with Pokey Player, Music Composer, and Music Construction Set. Whether you are a virtuoso musician or tone deaf, it doesn't matter. Our next meeting will be Wednesday, April 24th. The site of the meeting will be announced at tonight's MACE meeting. It will be posted on

the MACE BBSs or you can call me if you don't have a modem or can't make the MACE meeting (978-8432).

Our plans include the possible staging of a formal concert apart from the regular MACE meeting. We won't be able to do it, however, without more member support. If you are interested please call or attend the meeting!

OTHER SIGS

Assembler SIG: Meets 1st Thursday of each month. Contact Todd Meitzner at (313) 542-1752.

FORTH Interest Group: Contact Tom Chrapkiewicz at (313) 562-8506 or 845-4570 x60. For all FORTH users.

SIG-ED: Like a phoenix from its ashes, SIG-ED has risen. Call Chairman Mark Kennedy evenings at (313) 465-5849. (The April meeting will be on Wednesday, the 17th at Mt. Clemens High School.)

East Side SIG: Bad news for you lake-dwellers: the East Side SIG has disbanded.

SYSOP VIEWPOINT

by Mike Lechkun

[The following is the opinion of the MACE EAST BBS sysop, and in no way represents the opinion of MACE.]

So you just finished typing in that new BASIC game from the latest Atari-devoted magazine. You've run it through their program/typing check program, and everything seems to be in order. If you were smart, you saved it to disk or cassette. Otherwise you were like me: anxious to check it out and so you plunged right in. Then, whomp! The system locks up. Why this, after everything seemed to check out? Has the ghost of Jack Tramiel haunted your I.C.'s? Did the system whammy strike again?

Nope to all the above. You fell victim to an ad-man's plot. Ever wonder why the "Ooops!", "Excuse my errors", and "Corrections" pages in the mags are so big? That's to get you to buy the next month's magazine to find out what went wrong, what got omitted, and what was mis-typed from last month's programs.

I mean, c'mon now. You expect me to believe those are truly just accidents? There are many months' lead time involved before the issue you see in your hands is produced. The January issue of most major computer magazines is planned six months ahead in July, the deadline is in September, printing run in October, and it's on the shelves mid-November.

I have also talked to callers on the BBS who receive the "disk issue" of their favorite magazine. They claim the program is correct on the disk, but wrong in the mag. Just an accident, or intentional? If you feel as I do, then write the publishers of your favorite magazines and tell them to get with it and stop cheating us out of our $1.50 to $3.50 worth of magazine. There really is no excuse for it. These are professionals, and they treat the Atari owner for what they perceive him/her to be: just a game player.

I'll give some addresses of the magazines I feel are to blame for a lot of this stuff:

ANTIC
Jim Capparall, Editor
524 Second St.
San Fransisco, CA 94107

A.N.A.L.O.G.
Mike Deschenes, Editor
565 Main St.
Cherry Valley, MA 01611

COMPUTE!
Robert Lock, Editor in Chief
324 W. Wendover Ave.
Greensboro, NC 27408

Reader improvements in programs is one thing; blatant errors are another. Tell these people you want to see their magazines and the programs within them proofread completely before being issued to the public. We certainly deserve better treatment.

SOME DETROIT AREA ATARI BBSs

x = 24 hr $ = evening/nite B = 1200 bps capability

313-978-1685	MACE East	Bx
313-582-0657	MACE West (Secretary)	Bx
313-882-5909	Trading Post (President)	Bx
313-538-0197	Dartboard (Vice President)	x
313-771-4126	Freedom Board (Disk Librarian)	x
313-531-1701	Ethernet (Rec. Secretary)	Bx
313-641-8688	A.I.R. Port	x
313-978-8087	A.R.C.A.D.E.	x
313-278-3901	Atari Castle	Bx
313-525-5172	Baudville	Bx
313-546-3689	Bunky's Board	Bx
313-676-0696	Cougar's Lair	x
313-449-8544	Country Peddler	Bx
313-278-1727	Crazyhouse	Bx
313-291-3812	Cutting Board-MPP	$
313-565-6306	Dark World	x
313-264-1737	Enterprise	Bx
313-391-0474	H.A.C.C.	Bx
313-585-2168	Home Board-835	x
313-778-5279	Main Frame	x
313-368-4828	Playground	$
313-563-6177	Robot Factory	x
313-277-8632	Rock Palace	x
313-288-0824	Supership (formerly Superboard)	Bx
313-247-0094	Toolbox	Bx

313-545-8593	U.S.A.	x
313-532-8410	V.O.I.C.E.	x

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

SOME CANADIAN ATARI BBSs

416-488-6569	Dark World	Bx
519-473-9576	L.AT.C.H.	Bx
519-853-1063	Pitstop	Bx
416-622-7350	Starship	Bx

CASSETTE CORNER

by Mike Landis

Well, I'm sure you are wondering about the cassette corner. We have changed to the cassette, blank disk & submissions corner.

First off, we need more submissions. Your own software and articles are what MACE is all about. If you have an idea or a program, bring it back to my corner. We will help you along with your idea and get it into our own publications. We need your support, too!

Second, the response to the blank disks has been excellent! We are looking forward to bringing you more of the same excellent quality blank disks. Through your support we are able to buy in large quantity and get a good price for our memebers.

Third, there still are some cassettes left! We have priced them at $2.00 each. Even if you do not use them for your Atari, these tapes are premium quality, good for home taping! Note the low cost of these tapes; remember, these include poly boxes! For those of you who have been patiently waiting for the YA tape, it is here and it works! If there is a disk that you would like on cassette just let me know; you can special order any disk, except machine lanuage and OBJ files.

Starting next month I will continue my articles on the cassette recorder. For now, I will close with one last note: we recieved two submissions last meeting - let's break a record and have 4 new submissions next time! I know you have something, even a modification to a current program.

THE BBCS

by Kirk Revitzer
MACE President
Dedicated Sysop
of The Trading Post BBS
(313) 882-5909
24 hrs/7 days
300/1200 bps

Assuming you have a modem and also assuming that you use it to call the Trading Post you're probably wondering to yourself "What the heck is a BBCS?"

BBCS, The Bulletin Board Construction Set, is an Atari BBS written completely in machine language (using the Atari Macro-assembler (AMAC), I believe). The BBCS was written by Scott Brause of Edison, NJ and is now being "test run" in four locations, mine being one of them! By the way, Scott is Vice-President of the JACG user group and Sysop of the JACG BBCS.

So here's the number one item that catches the caller's eye: BBCS does NOT function in the ATASCII [Atari] mode. All of the functions, and text, are in regular ASCII mode. The reason is so that anyone (i.e. 1030 users) can go home and call in and not think something is wrong when the screen fills with "garbage". When transferring a file, if you're using Amodem with Xmodem protocol, it doesn't matter if you're in ATASCII or not. It will still work!

Anyhow, after over a week of running, the bugs are gone and it's running smooth. The board is password access (optional) or complete open access. Passwords allow for such things as private E-mail, etc. It has a very nice status window on the Sysop's end and any function of the board can be modified, added to, or deleted by the Sysop without taking the board down to do it!

Best yet, it's ready! BBCS, over a year in the making, is available now. It is not a "public domain" BBS program. There is an ad for it in the May '85 ANALOG Computing or you can call Scott at the JACG BBCS at 1-201-549-7591 for more info.

HAPPY BIRTHDAY MACE!

Believe it or not, MACE will be celebrating its 5th birthday at our May meeting. Mark your calendar now so you'll be sure not to miss it. There will be lots of door prizes (including a Grand Prize of an Atari 1050 disk drive - who couldn't use another one of those?), free refreshments (maybe we'll serve little blue peanuts in honor of the demise of the PC Jr), and...(keep your fingers crossed)...one of the new Atari XE machines.

Remember that this is the one meeting each year that is open to members only, so bring your card or your $20 or you won't get in.

Here are the meeting dates for the rest of '85:

6/18	7/23	8/20	9/17
10/15	11/19	12/17	

ATTENTION!

If you haven't reNEWed your MACE membership yet, STOP POKEing around. PEEK in your wallet or checkbook for $20.00, and PRINT on the handy form on the back page. RESTORE our faith in you by mailing your dues to the PO box or by paying at the next meeting. Please NOTE the date in the upper right hand corner of the mailing label; it is your membership expiration date. SAVE time and possible ERROR by reNEWing now. IF you have already taken care of this POINT, THEN thanks a lot. BYE.

TRAK DRIVE OWNERS:

Chuck Norris, Sysop of the Supership (formerly the Superboard), reports that he has found someone who can and will repair Trak drives! If yours isn't tracking right, contact:

Ernest Dodson
Microdoc, Inc.
Carleton, MI
(313) 654-9798

MACE JOURNAL LISTING CONVENTIONS

To reduce our readers' eyestrain, we have adopted a special method for listing programs. Programs will be listed in 38 column format, and certain characters will be replaced by an abbreviated form of their function, printed within curly braces (see below). Any characters to be typed in inverse video will be underlined, and control characters will be represented by their respective letters within curly braces. If a character within braces is also underlined, toggle the inverse video on and then hold down the control key while typing the character.

This method may seem awkward at first, but you should quickly get used to it, and the listings will be much easier to read. The special characters which will be spelled out are as follows:

When you see:	You should type:
{CLEAR}	ESC SHIFT <
{UP}	ESC CTRL -
{DOWN}	ESC CTRL =
{LEFT}	ESC CTRL +
{RIGHT}	ESC CTRL *
{BACK S}	ESC DELETE
{DELETE}	ESC CTRL DELETE
{INSERT}	ESC CTRL INSERT
{DEL LINE}	ESC SHIFT DELETE
{INS LINE}	ESC SHIFT INSERT
{TAB}	ESC TAB
{CLR TAB}	ESC CTRL TAB
{SET TAB}	ESC SHIFT TAB
{BELL}	ESC CTRL 2
{ESC}	ESC ESC
{COMMA}	CTRL , (comma)
{PERIOD}	CTRL . (period)
{SEMI-COLON}	CTRL ; (semi-colon)
{SHIFT =}	SHIFT =

If you see:	Type:
{A}	CTRL A
A	INV. VIDEO A
{A}	INV. VIDEO CTRL A

M. A. C. E.

MICHIGAN ATARI COMPUTER ENTHUSIASTS

P.O. Box 2785, Southfield, MI 48037

BULLETIN BOARDS: MACE 978-1685/MACE WEST 582-0657

MACE HOTLINE (Information - Voice Line) 882-7104

PRESIDENT
Kirk Revitzer
Detroit
882-9109
Compuserve: 70346,1642
BBS: 382-5909

VICE-PRESIDENT
Alva Thomas
Detroit
535-3748
Compuserve: 74065,334
BBS: 538-0197

TREASURER
Burt Gregory
Southfield
552-0273

PROGRAM COORDINATOR
Scott Garland
West Bloomfield
851-9453

CORRESPONDING SECRETARY
Sharie Middlebrook
Dearborn
581-5560
BBS: 582-0657

RECORDING SECRETARY
Dino Roggero
Redford
BBS: 531-1701

DISK LIBRARIAN
Dave Zappa
St. Clair Shores
773-8551
BBS: 771-4126

CASSETTE LIBRARIAN
Mike Landis
Clawson
589-1789
Compuserve: 72675,1023

MEMBERSHIP CHAIRMAN
Paul Wheeler
Detroit
538-3649

M.A.C.E. JOURNAL EDITOR
Ann McBain Ezzell
Birmingham
646-4455

NEXT MEETING: 5/21/85 7:00 PM
Southfield Pavilion--Ten & a Half Mile Road and Evergreen

M.A.C.E.
P.O Box 2785
Southfield, MI 48037

$20.00* FOR A ONE YEAR MEMBERSHIP
(*Payable to M.A.C.E.)

M.A.C.E. MEMBERSHIP APPLICATION

Name _____ Phone _____
Street _____
City _____ State _____ Zip Code _____
Company (if applicable) _____
System Description _____ Disk/Tape _____
Suggestions _____
I can help with . . . _____

( )New ( )Renewal
If a Renewal:
MACE #_____

Date _____ Coupon _____
Amount _____ Membership Card Number _____
☐ Cash Expires _____
☐ Check Number

MICHIGAN ATARI COMPUTER ENTHUSIASTS
P.O. BOX 2785
SOUTHFIELD, MICHIGAN 48037

IMPORTANT DATED MATERIAL

PLEASE DO NOT DELAY